

Time-space tradeoffs for two-way finite automata

Shenggen Zheng
zhengshenggen@gmail.com
Sun Yat-sen University

Daowen Qiu
issqdw@mail.sysu.edu.cn
Sun Yat-sen University

Jozef Gruska
gruska@fi.muni.cz
Masaryk University

Abstract

We explore bounds of *time-space tradeoffs* in language recognition on *two-way finite automata* for some special languages. We prove: (1) a time-space tradeoff upper bound for recognition of the languages $LEQ(n)$ on *two-way probabilistic finite automata* (2PFA): $TS = O(n \log n)$, whereas a time-space tradeoff lower bound on *two-way deterministic finite automata* is $\Omega(n^2)$; (2) a time-space tradeoff upper bound for recognition of the languages $L_{INT}(n)$ on *two-way finite automata with quantum and classical states* (2QCFA): $TS = O(n^{3/2} \log n)$, whereas a lower bound on 2PFA is $TS = \Omega(n^2)$; (3) a time-space tradeoff upper bound for recognition of the languages $L_{NE}(n)$ on exact 2QCFA: $TS = O(n^{1.87} \log n)$, whereas a lower bound on 2PFA is $TS = \Omega(n^2)$.

It has been proved (Klauck, STOC'00) that the exact one-way quantum finite automata have no advantage comparing to classical finite automata in recognizing languages. However, the result (3) shows that the exact 2QCFA do have an advantage in comparison with their classical counterparts, which has been the first example showing that the exact quantum computing have advantage in time-space tradeoff comparing to classical computing.

Usually, two communicating parties, Alice and Bob, are supposed to have an access to arbitrary computational power in *communication complexity* model that is used. Instead of that we will consider communication complexity in such a setting that two parties are using only finite automata and we prove in this setting that quantum automata are better than classical automata and also probabilistic automata are better than deterministic automata for some well known tasks.

1 Introduction

Time-space tradeoffs is an important research topic in the study of complexity of both classical and quantum computing [6, 19, 20] with respect to various computing models [7, 8, 12]. However, in the case of two-way finite automata, there is few work on their time-space tradeoffs. Mostly only their time complexity or state complexity (space complexity) has been investigated.

When just time complexity or state complexity (space complexity) of two-way finite automata to recognize some languages are considered, it seems that quantum finite automata have no advantages at all compared to their classical counterparts. However, quite surprisingly, when time-space product is considered, then advantages of quantum variations of the classical models can be demonstrated as shown in this paper.

Time-space tradeoffs are closely related to communication complexity. In this paper, we will use communication complexity results to derive time-space tradeoffs results for two-way finite automata. We prove that the time-space tradeoffs for recognizing some languages

Corresponding author: Daowen Qiu.

in two-way finite automata with quantum and classical states (2QCFA) [3] are better than in their classical counterparts and also that probabilistic two-way quantum finite automata (2PFA) [13] are better than two-way deterministic finite automata (2DFA) [17].

Since the topic of communication complexity was introduced by Yao [25], it has been extensively studied [10, 21]. In the setting of two parties, Alice is given an $x \in \{0, 1\}^n$, Bob is given a $y \in \{0, 1\}^n$ and their task is to communicate in order to determine the value of some given Boolean function $f : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}$, while exchanging as small number of bits as possible. In this setting, local computations of the parties are considered to be free, but communication is considered to be expensive and has to be minimized. Two of the most often studied communication problems are that of equality and intersection [21], defined as follows: (1) **Equality**: $\text{EQ}(x, y) = 1$ if $x = y$ and 0 otherwise. (2) **Intersection**: $\text{INT}(x, y) = 1$ if there is an index i such that $x_i = y_i = 1$ and 0 otherwise.

1.1 Time-space tradeoffs

Let us consider the following language over the alphabet $\Sigma = \{0, 1, \#\}$:

$$L_{\text{EQ}}(n) = \{x\#^ny \mid x, y \in \{0, 1\}^n, \text{EQ}(x, y) = 1\}. \quad (1)$$

It is clear that 2DFA (therefore also 2PFA) can recognize $L_{\text{EQ}}(n)$. The time complexity¹ of 2DFA recognizing this language is $\mathbf{O}(n)$. The state complexity of 2DFA recognizing the language is $\mathbf{O}(n^2)$, that is the space used is $\mathbf{O}(\log n)$. The time complexity and also the space used of 2PFA recognizing the same language is almost the same. However, when we consider time-space tradeoff for the language $L_{\text{EQ}}(n)$, the situation is very different.

We will use a 2PFA to simulate the probabilistic communication protocol from Chapter 1 of [16] for the problem EQ and get an upper bound for the time-space tradeoff for 2PFA.

Theorem 1. *There is a 2PFA that accepts the language $L_{\text{EQ}}(n)$ in the time T using the space S such that $TS = \mathbf{O}(n \log n)$.*

Using communication complexity lower bound proof method [21], we can get the lower bound for time-space tradeoff for 2DFA.

Theorem 2. *Let \mathcal{A} be a 2DFA that accepts the language $L_{\text{EQ}}(n)$ in time T using space S . Then, $TS = \Omega(n^2)$.*

In order to prove the time-space tradeoffs advantages of 2QCFA compared to 2PFA, let us consider the following language over the alphabet $\Sigma = \{0, 1, \#\}$:

$$L_{\text{INT}}(n) = \{x\#^ny \mid x, y \in \{0, 1\}^n, \text{INT}(x, y) = 1\}. \quad (2)$$

We use a 2QCFA to simulate the quantum communication protocol from [9, 10] for the problem INT and get an upper bound for the time-space tradeoff for 2QCFA.

Theorem 3. *There is a 2QCFA that accepts the language $L_{\text{INT}}(n)$ in time T using space S such that $TS = \mathbf{O}(n^{3/2} \log n)$.*

¹When two-way finite automata are used to recognize languages, they can halt before reading all the input.

Buhrman et al. [9] reduced certain quantum communication tasks to computation problems, which is essentially a way to transform quantum query algorithms to quantum communication protocols. More exactly, they showed that if there is a t -query quantum algorithm computing an n -bit Boolean function f with an error ε , then there is a communication protocol with $\mathbf{O}(t \log n)$ communication for the function $f(x \wedge y)$ with the same error ε .

The main idea in the proofs of our main results is to transform quantum query algorithms and quantum communication protocols to algorithms for 2QCFA.

Using one of communication complexity lower bound proof methods, we can get the following lower bound for the time-space tradeoff for the language $L_{INT}(n)$ on 2PFA.

Theorem 4. *Let \mathcal{A} be a 2PFA that accepts the language $L_{INT}(n)$ in time T using space S . Then, $TS = \Omega(n^2)$.*

Concerning the exact computing mode, Klauck [18] proved, for any regular language L , that the state complexity of the exact one-way quantum finite automata (1QFA) for L is not less than the state complexity of an equivalent one-way deterministic finite automata (DFA). That means that the exact 1QFA have no advantage in recognizing regular languages. It is therefore of interest to consider the case of two-way finite automata. We still do not know whether there is time complexity or state complexity advantages for two-way quantum finite automata in recognition of languages. However, we prove that exact 2QCFA do have time-space tradeoff advantages for recognizing some special languages.

Let us consider the sequence of functions studied in [5]. We define the function $NE(x_1, x_2, x_3)$ as follows: $NE(x_1, x_2, x_3) = 0$ if $x_1 = x_2 = x_3$ and $NE(x_1, x_2, x_3) = 1$ otherwise. Define

- $NE^0(x_1) = x_1$ and
- $NE^d(x_1, \dots, x_{3^d}) = NE(NE^{d-1}(x_1, \dots, x_{3^{d-1}}), NE^{d-1}(x_{3^{d-1}+1}, \dots, x_{2 \cdot 3^{d-1}}), NE^{d-1}(x_{2 \cdot 3^{d-1}+1}, \dots, x_{3^d}))$ for all $d > 0$.

Let $n = 3^d$. We define $RNE(x, y) = NE^d(x_1 \wedge y_1, \dots, x_n \wedge y_n)$, where $x, y \in \{0, 1\}^n$, and let us consider the following language

$$L_{NE}(n) = \{x \#^n y \mid x, y \in \{0, 1\}^n, RNE(x, y) = 1\}. \quad (3)$$

We will use a 2QCFA to simulate the quantum communication protocol from [5] for the problem RNE and get an upper bound for the time-space tradeoff for 2QCFA.

Theorem 5. *There is an exact 2QCFA that accepts the language $L_{NE}(n)$ in time T using space S such that $TS = \mathbf{O}(n^{1.87} \log n)$.*

Theorem 6. *Let \mathcal{A} be a 2PFA that accepts the language $L_{NE}(n)$ in time T using space S . Then, $TS = \Omega(n^2)$.*

1.2 Communication of finite automata

Two communicating parties Alice and Bob are usually supposed to have unlimited computational power in communication complexity models. However we will consider a very different setting. Namely that two parties are using only finite automata for their internal computation. In this setting, Alice and Bob will be sending only some states of their finite automata as messages to each other. At the beginning, Alice does some computation on her finite automaton, then sends a state s of her automaton to Bob. After receiving the state s from

Alice, Bob does computation with s as the starting state on his automaton and then after some computation Bob sends his state t to Alice. Alice then resumes computation in her automaton with the starting state t and so on. In case Alice and Bob are using 2QCFA, they can send both quantum and classical states.

We prove that the communication complexity for problems EQ, INT and RNE are almost the same as in the case both parties have unlimited computational power. Namely, we show:

Theorem 7. *The probabilistic communication complexity for EQ is $\mathbf{O}(\log n)$ when parties are using 2PFA.*

Theorem 8. *The quantum communication complexity for INT is $\mathbf{O}(\sqrt{n} \log n)$ when parties are using 2QCFA.*

Theorem 9. *The exact quantum communication complexity for RNE is $\mathbf{O}(n^{0.87} \log n)$ when parties are using 2QCFA.*

It seems that for many well known problems, changing the two communicating parties' computation power to finite automata only does not affect the communication complexity a lot.

2 Preliminaries

2.1 Quantum query algorithm

In the following let input $x = x_1 \cdots x_n \in \{0, 1\}^n$ for some fixed n . We will consider a Hilbert space \mathcal{H} with basis states $|i, j\rangle$ for $i \in \{0, 1, \dots, n\}$ and $j \in \{1, \dots, m\}$ (where m can be chosen arbitrarily). A query O_x to an input $x \in \{0, 1\}^n$ will be formulated as the following unitary transformation:

- $O_x|0, j\rangle = |0, j\rangle$;
- $O_x|i, j\rangle = (-1)^{x_i}|i, j\rangle$ for $i \in \{1, 2, \dots, n\}$.

A quantum query algorithm \mathcal{A} which uses t queries for an input x consists of a sequence of unitary operators $U_0, O_x, U_1, \dots, O_x, U_t$, where U_i 's do not depend on the input x and the query O_x does. The algorithm will start in a fixed starting state $|\psi_s\rangle$ of \mathcal{H} and will perform the above sequence of operations. This leads to the final state

$$|\psi_f\rangle = U_t O_x U_{t-1} \cdots U_1 O_x U_0 |\psi_s\rangle. \quad (4)$$

The final state is then measured with a measurement $\{M_0, M_1\}$. For an input $x \in \{0, 1\}^n$, we denote $\mathcal{A}(x)$ the output of the quantum query algorithm \mathcal{A} . Obviously, $Pr[\mathcal{A}(x) = 0] = \|M_0|\psi_f\rangle\|^2$ and $Pr[\mathcal{A}(x) = 1] = \|M_1|\psi_f\rangle\|^2 = 1 - Pr[\mathcal{A}(x) = 0]$. We say that the quantum query algorithm \mathcal{A} computes f within an error ε if for every input $x \in \{0, 1\}^n$ it holds that $Pr[\mathcal{A}(x) = f(x)] \geq 1 - \varepsilon$. If $\varepsilon = 0$, we say that the quantum algorithm is an exact quantum algorithm. For more details on the definition of quantum query complexity see [5, 11].

2.2 Communication complexity

We will use the following standard model of communication complexity. Two parties Alice and Bob compute a function f on distributed inputs x and y . A deterministic communication protocol \mathcal{P} will compute a function f , if for every input pair $(x, y) \in X \times Y$ the protocol terminates with the value $f(x, y)$ as its output at a well specified party. In a probabilistic protocol, Alice and Bob may also flip coins during the protocol execution and proceed according to outcomes of the coins. Moreover, the protocol can have an erroneous output with a small probability. In a quantum protocol, Alice and Bob may use also quantum resources for communication. Let $\mathcal{P}(x, y)$ denote the output of the protocol \mathcal{P} . We will consider two kinds of protocols for computing a function f :

- An exact protocol \mathcal{P} such that $\Pr(\mathcal{P}(x, y) = f(x, y)) = 1$.
- A bounded error protocol \mathcal{P} such that $\Pr(\mathcal{P}(x, y) = f(x, y)) \geq \frac{2}{3}$.

The communication complexity of a protocol \mathcal{P} is the number of (qu)bits exchanged in the worst case. The communication complexity of f is, with respect to the communication mode used, the complexity of an optimal protocol for f . We will use $D(f)$ and $R(f)$ to denote the deterministic communication complexity and the bounded error probabilistic communication complexity of the function f , respectively. Similarly, we use notations $Q_E(f)$ and $Q(f)$ for the exact and bounded error quantum communication complexity of a function f . For more details on the definition of communication complexity see [10, 21].

Some communication complexity results that we will use in this paper are:

1. $D(\text{EQ}) = \Omega(n)$, $R(\text{EQ}) = \mathcal{O}(\log n)$ [21].
2. $R(\text{INT}) = \Omega(n)$ [24], $Q(\text{INT}) = \mathcal{O}(\sqrt{n} \log n)$ [10].
3. $R(\text{RNE}) = \Omega(n)$, $Q_E(\text{RNE}) = \mathcal{O}(n^{0.87} \log n)$ [5].

2.3 Two-way finite automata

We assume familiarity with the models of finite automata introduced in [3, 13, 17]. We denote the input alphabet by Σ , which does not include symbols \clubsuit (the left end-marker) and $\$$ (the right end-marker). A two-way finite automaton that we will use in this paper halts when it enters an accepting or a rejecting state.

2QCFA were introduced by Ambainis and Watrous [3] and further studied by Zheng *et al.* [15, 22, 23, 26, 27, 28, 29, 30]. Informally, a 2QCFA can be seen as a 2DFA with an access to a quantum memory for states of a fixed Hilbert space upon which at each step either a unitary operation is performed or a projective measurement and the outcomes of which then probabilistically determine the next move of the underlying 2DFA.

A 2QCFA \mathcal{M} is specified by a 9-tuple

$$\mathcal{M} = (Q, S, \Sigma, \Theta, \delta, |q_0\rangle, s_0, S_{acc}, S_{rej}) \quad (5)$$

where:

1. Q is a finite set of orthonormal quantum basis states.
2. S is a finite set of classical states.
3. Σ is a finite alphabet of input symbols and let $\Sigma' = \Sigma \cup \{\clubsuit, \$\}$, where \clubsuit will be used as the left end-marker and $\$$ as the right end-marker.

4. $|q_0\rangle \in Q$ is the initial quantum state.
5. s_0 is the initial classical state.
6. $S_{acc} \subset S$ and $S_{rej} \subset S$, where $S_{acc} \cap S_{rej} = \emptyset$ are sets of the classical accepting and rejecting states, respectively.
7. Θ is a quantum transition function

$$\Theta : S \setminus (S_{acc} \cup S_{rej}) \times \Sigma' \rightarrow U(H(Q)) \cup O(H(Q)), \quad (6)$$

where $U(H(Q))$ and $O(H(Q))$ are sets of unitary operations and measurements on the Hilbert space generated by quantum states from Q .

8. δ is a classical transition function. If the automaton \mathcal{M} is in the classical state s , in the quantum state $|\psi\rangle$, and its tape head is scanning a symbol σ , then \mathcal{M} performs quantum and classical transitions as follows.

- (a) If $\Theta(s, \sigma) \in U(H(Q))$, then the unitary operation $\Theta(s, \sigma)$ is applied on the current quantum state $|\psi\rangle$ to produce a new quantum state. The automaton then performs, in addition, the following classical transition function

$$\delta : S \setminus (S_{acc} \cup S_{rej}) \times \Sigma' \rightarrow S \times \{-1, 0, 1\}. \quad (7)$$

If $\delta(s, \sigma) = (s', d)$, then the new classical state of the automaton will be s' and its head moves in the direction d .

- (b) If $\Theta(s, \sigma) \in O(H(Q))$, then the measurement operation $\Theta(s, \sigma)$ is applied on the current state $|\psi\rangle$. Suppose the measurement $\Theta(s, \sigma)$ is specified by operators $\{P_1, \dots, P_m\}$ and its corresponding classical outcome is from the set $N_{\Theta(s, \sigma)} = \{1, 2, \dots, m\}$. The classical transition function δ can be then specified as follow

$$\delta : S \setminus (S_{acc} \cup S_{rej}) \times \Sigma' \times N_{\Theta(s, \sigma)} \rightarrow S \times \{-1, 0, 1\}. \quad (8)$$

In such a case, if i is the classical outcome of the measurement, then the current quantum state $|\psi\rangle$ is changed to the state $P_i|\psi\rangle/\|P_i|\psi\rangle\|$. Moreover, if $\delta(s, \sigma)(i) = (s', d)$, then the new classical state of the automaton is s' and its head moves in the direction d .

The automaton halts and accepts (rejects) the input when it enters a classical accepting (rejecting) state (from $S_{acc}(S_{rej})$).

The computation of a 2QCFA $\mathcal{M} = (Q, S, \Sigma, \Theta, \delta, |q_0\rangle, s_0, S_{acc}, S_{rej})$ on an input $w \in \Sigma^*$ starts with the string $\$x\$$ on the input tape. At the start, the tape head of the automation is positioned on the left end-marker and the automaton begins the computation in the classical initial state s_0 and in the initial quantum state $|q_0\rangle$. After that, in each step, if its classical state is s , its tape head reads a symbol σ and its quantum state is $|\psi\rangle$, then the automaton changes its states and makes its head movement following the steps described in the definition.

Let $0 \leq \varepsilon < \frac{1}{3}$. A finite automaton \mathcal{M} recognizes a language L with error ε if, for $w \in \Sigma^*$,

1. $\forall w \in L, \Pr[\mathcal{M} \text{ accepts } w] \geq 1 - \varepsilon$, and
2. $\forall w \notin L, \Pr[\mathcal{M} \text{ rejects } w] \geq 1 - \varepsilon$.

If $\varepsilon = 0$, we say the finite automaton \mathcal{M} is an exact finite automaton.

3 Proofs

Proof of Theorem 1. We describe a 2PFA \mathcal{A} to accept the language $L_{EQ}(n)$. The automaton will use states $s_{q,k,l}$ where $0 \leq q, k, l \leq n^2$.

First of all, \mathcal{A} uses $O(n)$ states to check that the input is in the form $x\#^ny$, where $|x| = |y| = n$. If the length of the input $|w| > 3n$, then the automaton halts and rejects the input in $O(n)$ time. After that \mathcal{A} starts an addition computation in the state $s_{0,0,0}$. After reading the left-end marker, the automaton changes its state randomly to $s_{p,0,0}$, where $p \leq n^2$ is a prime. When the 2PFA \mathcal{A} reads the “ x -region”, it changes its state from $s_{p,0,0}$ to $s_{p,s,0}$, where $s = \text{Num}(x) \bmod p$. $\text{Num}(x)$ is the natural number whose binary representation is the string x . It is clear that such computation can be done by a 2PFA. When \mathcal{A} reads the “ $\#$ -region”, it keeps its state unchanged. When \mathcal{A} reads the “ y -region”, it changes its state from $s_{p,s,0}$ to $s_{p,s,t}$, where $t = \text{Num}(y) \bmod p$. The automaton reaches the right end-marker in a state $s_{p,s,t}$. If $s = t$, then the input is accepted. If $s \neq t$, the input is rejected.

\mathcal{A} actually simulates the communication protocol [16] for the problem EQ. If the input $w \in L_{EQ}(n)$, \mathcal{A} will accept it for certainty.

Let us now say that a prime $2 < p < n^2$ is bad for a pair (x, y) such that $x \neq y$, if the above 2PFA for such an input pair (x, y) and such a choice of prime yields a wrong answer. It is clear that there are at most $n - 1$ bad primes. Let $\text{Prime}(m)$ be the number of primes smaller than m . By the Prime number theorem, $\text{Prime}(n^2) > \frac{n^2}{2 \ln n}$.

If the input $x\#^ny \notin L_{EQ}(n)$, \mathcal{A} accepts the input only with the probability

$$\frac{\text{number of bad primes}}{\text{Prime}(n^2)} < \frac{n-1}{n^2/2 \ln n} < \frac{2 \ln n}{n}. \quad (9)$$

Obviously, the space used by \mathcal{A} is $S = O(\log n^6) = O(\log n)$ and the time is $T = O(n)$. Therefore, $TS = O(n \log n)$. \square

Proof of Theorem 2. Let \mathcal{A} be a 2DFA that recognizes the language $L_{EQ}(n)$ in time T using space S . We describe now a deterministic communication protocol for Alice and Bob that solves the problem EQ.

For an input $(x, y) \in \{0, 1\}^n \times \{0, 1\}^n$, Alice and Bob simulate \mathcal{A} with the input $x\#^ny$, where $x, y \in \{0, 1\}^n$. It is obvious that $x\#^ny \in L_{EQ}(n)$ iff $\text{EQ}(x, y) = 1$. Alice starts to simulate \mathcal{A} 's computation as long as the tape head of \mathcal{A} is either in “ x -region” of the input or in the “ $\#$ -region” of the input. When the tape head of \mathcal{A} moves to the “ y -region”, then Bob simulates \mathcal{A} 's computation as long as the tape head of \mathcal{A} is either in the “ y -region” of the input or in the “ $\#$ -region” of the input. When the tape head of \mathcal{A} moves to the “ x -region” of the input, Alice simulates \mathcal{A} 's computation again. The idea is that each player is responsible for the simulation in regions where he knows the input bits. In any step in which the tape goes from “ x -region” and “ $\#$ -region” to “ y -region” (from “ y -region” and “ $\#$ -region” to “ x -region”), Alice (Bob) sends the current state of \mathcal{A} to Bob (Alice).

In each time, the information which is required to send to the other party is not more than S . Since move from the “ x -region” to the “ y -region” and vice versa takes at least n steps (at least the size of “ $\#$ -region”), the number of times Alice and Bob send information to each other is at most T/n . All together the amount of communicating information in the protocol is not more than $S \cdot T/n$. Since $D(\text{EQ}) = \Omega(n)$ [21], we have $S \cdot T/n = \Omega(n)$ and therefore $TS = \Omega(n^2)$. \square

- Check that the input x is of the form of $\{0, 1\}^n$. Repeat the following ad infinity:
1. Read the left end-marker \clubsuit , perform $\Theta(s_0, \clubsuit)$ on the initial quantum state $|0\rangle$, change its classical state to $\delta(s_0, \clubsuit) = s_{1,1}$, and move the tape head one cell to the right.
 2. While the current classical state is not $s_{t+1,1}$, do the following
 - 2.1 While the currently scanned symbol σ is not the right end marker $\$$, do the following:
 - 2.1.1 Apply $\Theta(s_{k,i}, \sigma)$ to the current quantum state.
 - 2.1.2 Change the classical state $s_{k,i}$ to $s_{k,i+1}$ and move the tape head one cell to the right.
 - 2.2 When the right end-marker $\$$ is reached, perform $\Theta(s_{k,n+1}, \$) = U_k$ on the current quantum state. Change the classical state $s_{k,n+1}$ to $s_{k+1,1}$ and move the tape head to the symbol of x .
 3. Measure the current quantum state with the measurement $\{M_0, M_1\}$.
If the outcome is 1, the input is accepted. Otherwise, the input is rejected.

FIGURE 1: Description of the behavior of 2QCFA \mathcal{M} when simulating the quantum algorithm \mathcal{A} .

Before we prove Theorem 3, we present a main proof technique of this paper. Namely, that every quantum query algorithm can be simulated by a 2QCFA.

Theorem 10. *The computation of a quantum query algorithm \mathcal{A} for a Boolean function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ can be simulated by a 2QCFA \mathcal{M} . Moreover, if the quantum query algorithm \mathcal{A} uses t queries and l quantum basis states, then the 2QCFA \mathcal{M} uses $\mathbf{O}(l)$ quantum basis states, $\mathbf{O}(n^2)$ classical states, and $\mathbf{O}(t \cdot n)$ time.*

Proof. Suppose that the quantum query algorithm \mathcal{A} which use t queries is defined as in subsection 2.1. The input of the 2QCFA \mathcal{M} is the same as the input of the quantum query algorithm \mathcal{A} , which is $\clubsuit x \$$ on its tape. The main idea of the simulation goes as follows: We consider now a 2QCFA \mathcal{M} with quantum basis states $|0\rangle$ and $|i, j\rangle$ for $i \in \{0, 1, \dots, n\}$ and $j \in \{1, \dots, m\}$. \mathcal{M} starts its computation in the initial quantum state $|0\rangle$ and the initial classical state s_0 . The first time when \mathcal{M} reads the left-end marker \clubsuit , \mathcal{M} applies $\Theta(s_0, \clubsuit)$ to the quantum state such that $\Theta(s_0, \clubsuit)|0\rangle = U_0|\psi_s\rangle$.

The k -th time when \mathcal{M} reads the right-end marker $\$$, \mathcal{M} applies U_k to the quantum state, where $1 \leq k \leq t$. \mathcal{M} simulates the query O_x every time when it reads the input $x = x_1 \cdots x_n$ from left to right. The automaton proceeds precisely as in Figure 1, where

$$\Theta(s_{k,i}, \sigma)|0\rangle = |0\rangle, \quad \Theta(s_{k,i}, \sigma)|i, j\rangle = (-1)^\sigma |i, j\rangle \quad \text{and} \quad \Theta(s_{k,i}, \sigma)|u, j\rangle = |u, j\rangle \quad \text{for } u \neq i. \quad (10)$$

It is easy to verify that the unitary operators preformed in Step 2.1 are

$$\Theta(s_{k,n}, x_n)\Theta(s_{k,n-1}, x_{n-1}) \dots \Theta(s_{k,1}, x_1) = O_x. \quad (11)$$

It is clear that for any input x , $Pr[\mathcal{A}(x) = 1] = Pr[\mathcal{M} \text{ accepts } x]$ and $Pr[\mathcal{A}(x) = 0] = Pr[\mathcal{M} \text{ rejects } x]$. From the above simulation, we can see that if the quantum query algorithm \mathcal{A} uses l quantum basis states and t queries, then the 2QCFA \mathcal{M} uses $\mathbf{O}(l)$ quantum basis states, $\mathbf{O}(n^2)$ classical states, and $\mathbf{O}(t \cdot n)$ time. \square

We have proved that 2QCFA can simulate quantum query algorithms. Now what about the quantum communication protocol for the INT problem? According to [9, 10], we need to simulate the following unitary map:

$$O_z : |i\rangle \mapsto (-1)^{z_i} |i\rangle, \quad (12)$$

1. Move the tape head to the first symbol of x , set its classical state to s_1 .
2. While the currently scanned symbol σ is not $\#$, do the following:
 - 2.1 Apply $\Theta(s_i, \sigma) = U_{i, \sigma}$ to the current quantum state.
 - 2.2 Change the classical state s_i to s_{i+1} and move the tape head one cell to the right.
3. Move the tape head to the first symbol of y .
4. While the currently scanned symbol σ is not $\$$, do the following:
 - 4.1 Apply $\Theta(s_{n+i}, \sigma) = V_{i, \sigma}$ to the current quantum state.
 - 4.2 Change the classical state s_{n+i} to s_{n+i+1} and move the tape head one cell to the right.
5. Change the classical state s_{2n+1} to s_1 and move the tape head to the first symbol of x .
6. While the currently scanned symbol σ is not $\#$, do the following:
 - 6.1 Apply $\Theta(s_i, \sigma) = U_{i, \sigma}$ to the current quantum state.
 - 6.2 Change the classical state s_i to s_{i+1} and move the tape head one cell to the right.

FIGURE 2: Description of the behavior of 2QCFA when simulating the unitary map O_z .

where $z = x \wedge y$ is a bit-wise AND of x and y , since $z_i = 1$ whenever both $x_i = 1$ and $y_i = 1$.

Lemma 1. *Let $w = x\#^ny$, where $x, y \in \{0, 1\}^n$, be the input of a 2QCFA \mathcal{M} . Then, the unitary map: $O_z : |i\rangle \mapsto (-1)^{z_i}|i\rangle$, where $z = x \wedge y$, can be simulated by \mathcal{M} . Moreover, \mathcal{M} uses one additional auxiliary qubit and $\mathbf{O}(n)$ classical states and its running time is $\mathbf{O}(n)$.*

Proof. Assume that Alice wants to apply O_z to a quantum state $|\phi\rangle = \sum_{i=1}^n \alpha_i |i\rangle$. \mathcal{M} will use quantum states $\{|i\rangle|0\rangle, |i\rangle|1\rangle\}_{i=1}^n$ and classical states $\{s_i\}_{i=0}^{2n+1}$. \mathcal{M} will start with the quantum state $|\phi\rangle|0\rangle$. The procedure to simulate the unitary map O_z is as in Figure 2, where

$$U_{i, \sigma} |j\rangle |b\rangle = |j\rangle |b \oplus \sigma\rangle \text{ if } j = i, \text{ otherwise } U_{i, \sigma} |j\rangle |b\rangle = |j\rangle |b\rangle; \quad (13)$$

$$V_{i, \sigma} |j\rangle |1\rangle = (-1)^\sigma |j\rangle |1\rangle \text{ if } j = i, \text{ otherwise } V_{i, \sigma} |j\rangle |b\rangle = |j\rangle |b\rangle. \quad (14)$$

It is easy to verify that $U_{i, \sigma}$ and $V_{i, \sigma}$ are unitary. After Step 2, the quantum state changes to

$$U_{n, x_n} \cdots U_{1, x_1} \sum_{i=1}^n \alpha_i |i\rangle |0\rangle = \sum_{i=1}^n \alpha_i |i\rangle |x_i\rangle. \quad (15)$$

After Step 4, the quantum state changes to

$$V_{n, y_n} \cdots V_{1, y_1} \sum_{i=1}^n \alpha_i |i\rangle |x_i\rangle = \sum_{i=1}^n \alpha_i \cdot (-1)^{x_i \wedge y_i} |i\rangle |x_i\rangle. \quad (16)$$

After Step 6, the quantum state changes to

$$U_{n, x_n} \cdots U_{1, x_1} \sum_{i=1}^n \alpha_i \cdot (-1)^{x_i \wedge y_i} |i\rangle |x_i\rangle = \sum_{i=1}^n \alpha_i \cdot (-1)^{x_i \wedge y_i} |i\rangle |0\rangle = O_z |\phi\rangle |0\rangle. \quad (17)$$

□

Proof of Theorem 3. Combining the simulation techniques from Theorem 10 and Lemma 1, we can use a 2QCFA to simulate a Grover search [14] on the input $z \in \{0, 1\}^n$, where $z_i = x_i \wedge y_i$. Therefore it is clear that there is a 2QCFA recognizing the language $L_{INT}(n)$. Since the Grover's algorithm requires $\mathbf{O}(\sqrt{n})$ queries and uses $\mathbf{O}(n)$ quantum basis states, the

time used by the 2QCFA is $T = \mathbf{O}(\sqrt{n} \cdot n) = \mathbf{O}(n^{3/2})$. The number of quantum states used by the 2QCFA is $\mathbf{O}(n)$ and the number of classical states is $\mathbf{O}(n^2)$. Therefore, the space used by the 2QCFA is $S = \mathbf{O}(\log n + \log n^2) = \mathbf{O}(\log n)$. Hence, $TS = \mathbf{O}(n^{3/2} \log n)$. \square

Proof of Theorem 4. Similar to the proof of Theorem 2 except that probabilistic computation is used instead of deterministic one. The result is based on $R(\text{INT}) = \Omega(n)$ [24]. \square

Proof of Theorem 5. Combining the simulation techniques from Theorem 10 and Lemma 1, we can use a 2QCFA to simulate Ambainis' exact query algorithm in [5] on the input $z \in \{0, 1\}^n$, where $z_i = x_i \wedge y_i$. Therefore, there is an exact 2QCFA recognizing the language $L_{NE}(n)$. Since the exact algorithm requires $\mathbf{O}(n^{0.87})$ queries and uses $\mathbf{O}(n)$ quantum basis states, the time used by the exact 2QCFA is $T = \mathbf{O}(n^{0.87} \cdot n) = \mathbf{O}(n^{1.87})$. The space used is $S = \mathbf{O}(\log n)$. Hence, $TS = \mathbf{O}(n^{1.87} \log n)$. \square

Proof of Theorem 6. The proof is similar to that of Theorem 2 except that probabilistic computation is used instead of deterministic one. The final result is then based on $R(\text{RNE}) = \Omega(n)$ [5]. \square

For the proofs of Theorems 7, 8 and 9, it is clear that the communication protocols can be picked up from the proofs of Theorems 1, 3 and 5, respectively. We omit the details of proofs here.

4 Conclusion and open problems

Query complexity and communication complexity are related to each other. By using a simulation technique that transforms quantum query algorithms to quantum communication protocols, Buhrman et al. [9, 10] obtained new quantum communication protocols and showed the first exponential gap between quantum and classical communication complexity.

In this paper, we have developed the connection among 2QCFA, quantum communication protocols and quantum query algorithms. We have constructed 2QCFA to simulate quantum query algorithms. Using known quantum query algorithms and quantum communication protocols, this simulation enabled us to prove several time-space tradeoff results for 2QCFA. It also enabled us to find out communication protocols for the case that two parties are using 2QCFA for computation. It is clear that if the protocol is a one-way communication protocol, then the result can be directly transformed to the state complexity result of finite automata. For example, since the protocol in Theorem 7 is a one-way protocol, it is clear that the following result holds: the space complexity of one-way probability finite automata for the language $\{x\#y \mid x, y \in \{0, 1\}^n, \text{EQ}(x, y) = 1\}$ is $\mathbf{O}(\log n)$, whereas the space complexity for DFA is $\Omega(n)$. If a one-way quantum communication protocol is transformed from a quantum query complexity, then it can be implemented on 1QCFA and the space complexity result will follow immediately. Since we can use known results in quantum query complexity and communication complexity to derive new state succinctness results of quantum finite automata, the method is more general than the one used in [2].

Some problems for future research:

1. The quantum communication complexity tight bound $Q(\text{DISJ}) = \Theta(\sqrt{n})$ [1]. Does there exists a 2QCFA that accepts the language $L_{INT}(n)$ in time T using space S such that $TS = \mathbf{O}(n^{3/2})$?

2. We have shown, for the first time, that the time-space tradeoff TS on exact 2QCFA is superlinearly better than that for 2PFA in recognition of the language $L_{NE}(n)$. Can we find out more languages that 2QCFA have superlinear advantage? Find more examples such that exact quantum computing have superlinear advantage in time-space tradeoff for total functions in other computing models?
3. We have proved that the exact 2QCFA have superlinear advantage in time-space trade-off. Can we prove that exact 2QCFA have superlinear advantage in time complexity or space complexity in recognizing languages comparing to 2DFA or 2PFA?
4. We have transformed quantum computing advantages in communication complexity and query complexity to quantum finite automata. Can we do the opposite way? For instant, Ambainis and Freivalds [2] constructed a quantum finite automaton that is exponentially smaller than equivalent classical automaton, can we transform the problem to communication problem and prove the quantum communication complexity advantage?

Acknowledgements

S.G. Zheng thanks A. Ambainis, C. Mereghetti, B. Palano and L. Li for discussions.

References

- [1] Aaronson, S., and A. Ambainis, Quantum search of spatial regions, In Proceedings of 44th FOCS (2003) 200–209.
- [2] A. Ambainis and R. Freivalds, One-way quantum finite automata: strengths, weaknesses and generalizations, In Proceedings of the 39th FOCS (1998) 332–341.
- [3] A. Ambainis and J. Watrous, Two-way finite automata with quantum and classical states, TCS **287** (2002) 299–311.
- [4] A. Ambainis, A. Nayak, A. Ta-Shma and U. Vazirani, Dense quantum coding and quantum finite automata, Journal of the ACM, **49** (2002) 496–511.
- [5] A. Ambainis, Superlinear advantage for exact quantum algorithms, In Proceedings of 45th STOC (2013) 891–900.
- [6] A. Borodin and S. Cook, A time-space tradeoff for sorting on a general sequential model of computation, SIAM Journal on Computing **11** (1982) 287–297.
- [7] P. Beame, M. Saks and J. Thathachar, Time-space tradeoffs for branching programs, in Proceedings of 39th FOCS (1998) 254–263.
- [8] L. Babai, N. Nisan and M. Szegedy, Multiparty protocols, pseudorandom generators for logspace, and time-space trade-offs, JCSS **45** (1992) 204–232.
- [9] H. Buhrman, R. Cleve and A. Wigderson, Quantum vs. classical communication and computation, In Proceedings of 30th STOC (1998) 63–68.
- [10] H. Buhrman, R. Cleve, S. Massar and R. de Wolf, Nonlocality and Communication Complexity, Rev. Mod. Phys. **82** (2010) 665–698. Also arXiv:0907.3584 (2009).

- [11] H. Buhrman and R. de Wolf, Complexity measures and decision tree complexity: a survey, *TCS* **288** (2002) 21–43.
- [12] A. Cobham, The recognition problem for the set of perfect squares, *IEEE Conference Record Seventh SWAT* (1966) 78–87.
- [13] R. Freivalds, Probabilistic two-way machines, In *MFCS'81*, LNCS **188** (1981) 33–45.
- [14] L. K. Grover, A fast quantum mechanical algorithm for database search, In *Proceedings of 28th STOC* (1996) 212–219.
- [15] J. Gruska, D.W. Qiu, S.G. Zheng, Generalizations of the distributed Deutsch-Jozsa promise problem, *Mathematical Structures in Computer Science*, doi:10.1017/S0960129515000158, (2015). Also arXiv:1402.7254 (21 pages).
- [16] J. Hromkovič, *Design and Analysis of Randomized Algorithms*, Springer (2005).
- [17] J.E. Hopcroft and J.D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley, New York, (1979).
- [18] H. Klauck, On quantum and probabilistic communication: Las Vegas and one-way protocols, In *Proceedings of the 32th STOC*(2000) 644–651.
- [19] H. Klauck, Quantum time-space tradeoffs for sorting, In *Proceedings of 35th STOC* (2003) 69–76.
- [20] H. Klauck, R. Špalek and R. de Wolf, Quantum and classical strong direct product theorems and optimal time-space tradeoffs, *SIAM J. Comput.* **36** (2007) 1472–1493.
- [21] E. Kushilevitz, N. Nisan, *Communication Complexity*, Cambridge University Press (1997).
- [22] L.Z. Li, Y. Feng, On hybrid models of quantum finite automata, *Journal of Computer and System Science* **81** (2015) 1144–1158. Also arXiv:1206.2131.
- [23] D.W. Qiu, L. Li, P. Mateus, A. Sernadas, Exponentially more concise quantum recognition of non-RMM languages, *Journal of Computer and System Sciences* **81** (2) (2015) 359–375. Also arXiv:0909.1428.
- [24] A. Razborov, On the distributional complexity of disjointness, *TCS* **106** (1992) 385–390.
- [25] A. C. Yao, Some Complexity Questions Related to Distributed Computing, In *Proceedings of 11th STOC* (1979) 209–213.
- [26] A. Yakaryilmaz and A.C.C. Say, Succinctness of two-way probabilistic and quantum finite automata, *Discrete Mathematics and Theoretical Computer Science* **12** (2010) 19–40.
- [27] S.G. Zheng, D.W. Qiu, L.Z. Li and J. Gruska, One-way finite automata with quantum and classical states, *Dassow Festschrift*, LNCS **7300** (2012) 273–290. Also arXiv:1112.2022.

- [28] S.G. Zheng, D.W. Qiu, J. Gruska, L.Z. Li and P. Mateus, State succinctness of two-way finite automata with quantum and classical states, *Theoretical Computer Science* **499** (2013) 98–112. Also arXiv:1202.2651.
- [29] S.G. Zheng, J. Gruska, D.W. Qiu, On the state complexity of semi-quantum finite automata, *RAIRO-Theoretical Informatics and Applications* **48** (2014) 187–207. Earlier version in LATA’14. Also arXiv:1307.2499.
- [30] S.G. Zheng, Q.W. Qiu, J. Gruska, Power of the interactive proof systems with verifiers modeled by semi-quantum two-way finite automata, *Information and Computation* **241** (2015) 194–214. Also arXiv:1304.3876.